IEEE Xplore® RELEASE 2.4

Home | Login | Logout | Access Information | Alerts | Purchase History | Cart |

Welcome United States Patent and Trademark Office

□ Search Results

BROWSE SEARCH IEEE XPLORE GUIDE

Results for **"(((spam<in>metadata) <and> (url<in>metadata))<and> (detect<in>m..."**

e-mail

Your search matched **0** documents.
A maximum of **100** results are displayed, **25** to a page, sorted by **Relevance** in **Descending** order.

» Search Options

View Session History

New Search

Modify Search

(((spam<in>metadata) <and> (url<in>metadata))<and> (detect<in>metadata)) Search

□ Check to search only within this results set

Display Format: ● Citation ○ Citation & Abstract

» Key

IEEE JNL	IEEE Journal or Magazine
IET JNL	IET Journal or Magazine
IEEE CNF	IEEE Conference Proceeding
IET CNF	IET Conference Proceeding
IEEE STD	IEEE Standard

No results were found.

Please edit your search criteria and try again. Refer to the Help pages if you need assistan search.

Help Contact Us Privacy &

Indexed by Inspec®

EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
S1	14171456	@ad<"20040120"	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/09/29 13:24
S2	4854	709/206.ccls.	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/09/29 13:24
S3	3317	S1 and S2	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/09/29 13:24
S4	2459	spam and detect$	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/09/29 13:24
S5	3148	url with extract$	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/09/29 13:25
S6	104	S4 and S5	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/09/29 13:25
S7	6	S3 and S6	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/09/29 13:27
S8	6	hasegawa.in. and spam	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/09/29 13:27
S9	5	S8 not S7	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/09/29 13:27
S10	3	S1 and S9	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/09/29 13:27
S11	0	S2 and S10	US-PGPUB; USPAT; EPO; JPO; IBM_TDB	OR	ON	2007/09/29 13:27

PORTAL

USPTO

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

+spam +url extract detect analyze analysis SEARCH

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Published before February 2004
Terms used: **spam url extract detect analyze analysis**

Found **67** of **152,240**

Sort results by	relevance	
Display results	expanded form	

Save results to a Binder
Search Tips
□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 67

Result page: **1** 2 3 4 next

Relevance scale □ ▭ ▬ ■ ■

1 Fast detection of communication patterns in distributed executions ■
Thomas Kunz, Michiel F. H. Seuren
November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research CASCON '97**
Publisher: IBM Press
Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

2 Emergent web patterns: The connectivity sonar: detecting site functionality by structural patterns ■
Einat Amitay, David Carmel, Adam Darlow, Ronny Lempel, Aya Soffer
August 2003 **Proceedings of the fourteenth ACM conference on Hypertext and hypermedia HYPERTEXT '03**
Publisher: ACM Press
Full text available: pdf(153.40 KB) Additional Information: full citation, abstract, references, citings, index terms

Web sites today serve many different functions, such as corporate sites, search engines, e-stores, and so forth. As sites are created for different purposes, their structure and connectivity characteristics vary. However, this research argues that sites of similar role exhibit similar structural patterns, as the functionality of a site naturally induces a typical hyperlinked structure and typical connectivity patterns to and from the rest of the Web. Thus, the functionality of Web sites is refle ...

Keywords: link analysis, web IR, web graphs

3 PicASHOW: pictorial authority search by hyperlinks on the web ■
January 2002 **ACM Transactions on Information Systems (TOIS)**, Volume 20 Issue 1
Publisher: ACM Press
Full text available: Additional Information: full citation, abstract, references, index terms,

pdf(436.32 KB) review

We describe PicASHOW, a fully automated WWW image retrieval system that is based on several link-structure analyzing algorithms. Our basic premise is that a page *p* displays (or links to) an image when the author of *p* considers the image to be of value to the viewers of the page. We thus extend some well known link-based WWW *page retrieval* schemes to the context of image retrieval.PicASHOW's analysis of the link structure enables it to retrieve relevant images even when those ...

Keywords: Image retrieval, hubs and authorities, image hubs, link structure analysis

4 Information retrieval on the web
Mei Kobayashi, Koichi Takeda
June 2000 **ACM Computing Surveys (CSUR)**, Volume 32 Issue 2
Publisher: ACM Press
Full text available: pdf(213.89 KB) Additional Information: full citation, abstract, references, citings, index terms

In this paper we review studies of the growth of the Internet and technologies that are useful for information search and retrieval on the Web. We present data on the Internet from several different sources, e.g., current as well as projected number of users, hosts, and Web sites. Although numerical figures vary, overall trends cited by the sources are consistent and point to exponential growth in the past and in the coming decade. Hence it is not surprising that about 85% of Internet user ...

Keywords: Internet, World Wide Web, clustering, indexing, information retrieval, knowledge management, search engine

5 Web crawling and measurement: A large-scale study of the evolution of web pages
Dennis Fetterly, Mark Manasse, Marc Najork, Janet Wiener
May 2003 **Proceedings of the 12th international conference on World Wide Web WWW '03**
Publisher: ACM Press
Full text available: pdf(806.78 KB) Additional Information: full citation, abstract, references, citings, index terms

How fast does the web change? Does most of the content remain unchanged once it has been authored, or are the documents continuously updated? Do pages change a little or a lot? Is the extent of change correlated to any other property of the page? All of these questions are of interest to those who mine the web, including all the popular search engines, but few studies have been performed to date to answer them.One notable exception is a study by Cho and Garcia-Molina, who crawled a set of 720,00 ...

Keywords: degree of change, rate of change, web characterization, web evolution, web pages

6 Papers from Hotnets-II: The dark side of the Web: an open proxy's view
Vivek S. Pai, Limin Wang, KyoungSoo Park, Ruoming Pang, Larry Peterson
January 2004 **ACM SIGCOMM Computer Communication Review**, Volume 34 Issue 1
Publisher: ACM Press
Full text available: pdf(102.49 KB) Additional Information: full citation, abstract, references

With the advent of large-scale, wide-area networking testbeds, researchers can deploy long-running services that interact with other resources on the Web. While such interaction can easily attract clients and traffic, our experience suggests that projects accepting outside input and interacting with outside resources must carefully consider the avenues for abuse of such services. The CoDeeN Content Distribution Network, deployed on PlanetLab, uses a network of caching Web proxy servers to intell ...

7 Securing information: Guarding the next Internet frontier: countering denial of information attacks
Mustaque Ahamad, Leo Mark, Wenke Lee, Edward Omicienski, Andre dos Santos, Ling Liu, Calton Pu
September 2002 **Proceedings of the 2002 workshop on New security paradigms NSPW '02**
Publisher: ACM Press
Full text available: pdf(918.49 KB) Additional Information: full citation, abstract, references, citings, index terms

As applications enabled by the Internet become information rich, ensuring access to quality information in the presence of potentially malicious entities will be a major challenge. Denial of information (DoI) attacks attempt to degrade the quality of information by deliberately introducing noise that appears to be useful information. The mere availability of information is insufficient if the user must find a needle in a haystack of noise that is created by an adversary to hide critical informat ...

Keywords: countering information attacks, quality of information

8 Social browsing: Group unified histories an instrument for productive unconstrained co-browsing
Maria Aneiros, Vladimir Estivill-Castro, Chengzheng Sun
November 2003 **Proceedings of the 2003 international ACM SIGGROUP conference on Supporting group work GROUP '03**
Publisher: ACM Press
Full text available: pdf(223.25 KB) Additional Information: full citation, abstract, references, citings, index terms

The most common task being performed on the World Wide Web, namely exploring its contents remains an individual rather than a cooperative, shared or partnered activity. We propose that the existing model of collaborative browsing, namely master/slave, is too restrictive. Instead, we introduce group unified histories to provide unconstrained cooperative browsing. Our approach is founded on a persistent shared history object which is replicated for each user and totally configurable. In order for ...

Keywords: awareness, collaborative browsing, consistency model, group unified history, unconstrained cooperative browsing

9 SALSA: the stochastic approach for link-structure analysis
R. Lempel, S. Moran
April 2001 **ACM Transactions on Information Systems (TOIS)**, Volume 19 Issue 2
Publisher: ACM Press
Full text available: pdf(180.81 KB) Additional Information: full citation, abstract, references, citings, index terms

Today, when searching for information on the WWW, one usually performs a query through a term-based search engine. These engines return, as the query's result, a list of Web pages whose contents matches the query. For broad-topic queries, such searches often result in a huge set of retrieved documents, many of which are irrelevant to the user. However, much information is contained in the link-structure of the WWW.

Information such as which pages are linked to others can be used to augment searc ...

Keywords: Link-structure analysis, SALSA, TKC effect, hubs and authorities, random walks

10 The form is the substance: classification of genres in text
Nigel Dewdney, Carol VanEss-Dykema, Richard MacMillan
July 2001 **Proceedings of the workshop on Human Language Technology and Knowledge Management - Volume 2001**
Publisher: Association for Computational Linguistics
Full text available: pdf(64.60 KB) Additional Information: full citation, abstract, references

Categorization of text in IR has traditionally focused on topic. As use of the Internet and e-mail increases, categorization has become a key area of research as users demand methods of prioritizing documents. This work investigates text classification by format style, i.e. "genre", and demonstrates, by complementing topic classification, that it can significantly improve retrieval of information. The paper compares use of presentation features to word features, and the combination thereof, usin ...

11 Performance and cost tradeoffs in Web search
Nick Craswell, Francis Crimmins, David Hawking, Alistair Moffat
January 2004 **Proceedings of the 15th Australasian database conference - Volume 27 ADC '04**
Publisher: Australian Computer Society, Inc.
Full text available: pdf(153.92 KB) Additional Information: full citation, abstract, references, citings, index terms

Web search engines crawl the web to fetch the data that they index. In this paper we re-examine that need, and evaluate the network costs associated with data acquisition, and alternative ways in which a search service might be supported. As a concrete example, we make use of the Research Finder search service provided at http://rf.panopticsearch.com, and information derived from its crawl and query logs. Based upon an analysis of the Research Finder system we introduce a hybrid arrangement, in ...

Keywords: Web crawling, World-Wide Web, information retrieval, metasearch, search engine

12 Search 2: Evaluating strategies for similarity search on the web
Taher H. Haveliwala, Aristides Gionis, Dan Klein, Piotr Indyk
May 2002 **Proceedings of the 11th international conference on World Wide Web WWW '02**
Publisher: ACM Press
Full text available: pdf(268.54 KB) Additional Information: full citation, abstract, references, citings, index terms

Finding pages on the Web that are similar to a query page (Related Pages) is an important component of modern search engines. A variety of strategies have been proposed for answering Related Pages queries, but comparative evaluation by user studies is expensive, especially when large strategy spaces must be searched (e.g., when tuning parameters). We present a technique for automatically evaluating strategies using Web hierarchies, such as Open Directory, in place of user feedback. We apply this ...

Keywords: evaluation, open directory project, related pages, search, similarity search

13 Link-based ranking 2: Searching the workplace web

Ronald Fagin, Ravi Kumar, Kevin S. McCurley, Jasmine Novak, D. Sivakumar, John A. Tomlin, David P. Williamson

May 2003 **Proceedings of the 12th international conference on World Wide Web WWW '03**

Publisher: ACM Press

Full text available: pdf(231.55 KB) Additional Information: full citation, abstract, references, citings, index terms

The social impact from the World Wide Web cannot be underestimated, but technologies used to build the Web are also revolutionizing the sharing of business and government information within intranets. In many ways the lessons learned from the Internet carry over directly to intranets, but others do not apply. In particular, the social forces that guide the development of intranets are quite different, and the determination of a "good answer" for intranet search is quite different than on the Int ...

14 Integrating the document object model with hyperlinks for enhanced topic distillation and information extraction

Soumen Chakrabarti

April 2001 **Proceedings of the 10th international conference on World Wide Web WWW '01**

Publisher: ACM Press

Full text available: pdf(369.63 KB) Additional Information: full citation, references, citings, index terms

Keywords: document object model, minimum description length principle, segmentation, topic distillation

15 Query-independent evidence in home page finding

Trystan Upstill, Nick Craswell, David Hawking

July 2003 **ACM Transactions on Information Systems (TOIS)**, Volume 21 Issue 3

Publisher: ACM Press

Full text available: pdf(258.07 KB) Additional Information: full citation, abstract, references, citings, index terms

Hyperlink recommendation evidence, that is, evidence based on the structure of a web's link graph, is widely exploited by commercial Web search systems. However there is little published work to support its popularity. Another form of query-independent evidence, URL-type, has been shown to be beneficial on a home page finding task. We compared the usefulness of these types of evidence on the home page finding task, combined with both content and anchor text baselines. Our experiments made use of ...

Keywords: Web information retrieval, citation and link analysis, connectivity

16 PicASHOW: pictorial authority search by hyperlinks on the Web

Ronny Lempel, Aya Soffer

April 2001 **Proceedings of the 10th international conference on World Wide Web WWW '01**

Publisher: ACM Press

Full text available: pdf(633.77 KB) Additional Information: full citation, references, citings, index terms

Keywords: hubs and authorities, image hubs, image retrieval, link structure analysis

17 Automatically summarising Web sites: is there a way around it?
Einat Amitay, Cécile Paris
November 2000 **Proceedings of the ninth international conference on Information and knowledge management CIKM '00**
Publisher: ACM Press
Full text available: pdf(118.38 KB) Additional Information: full citation, references, citings, index terms

Keywords: Web site summarisation, information retrieval from links

18 Columns: Risks to the public in computers and related systems
Peter G. Neumann
January 2001 **ACM SIGSOFT Software Engineering Notes**, Volume 26 Issue 1
Publisher: ACM Press
Full text available: pdf(3.24 MB) Additional Information: full citation

19 Rank aggregation methods for the Web
Cynthia Dwork, Ravi Kumar, Moni Naor, D. Sivakumar
April 2001 **Proceedings of the 10th international conference on World Wide Web WWW '01**
Publisher: ACM Press
Full text available: pdf(288.25 KB) Additional Information: full citation, references, citings, index terms

Keywords: metasearch, multi-word queries, rank aggregation, ranking functions, spam

20 Risks to the public: Risks to the public in computers and related systems
Peter G. Neumann
May 2002 **ACM SIGSOFT Software Engineering Notes**, Volume 27 Issue 3
Publisher: ACM Press
Full text available: pdf(1.92 MB) Additional Information: full citation

Results 1 - 20 of 67 Result page: **1** 2 3 4 next

Useful downloads: Adobe Acrobat QuickTime Windows Media Player Real Player

An introduction to the Spambayes project

Full text Html (16 KB)

Source **Linux Journal** archive
Volume 2003 , Issue 107 (March 2003) table of contents
Page: 2
Year of Publication: 2003
ISSN:1075-3583

Author Richie Hindle

Publisher Specialized Systems Consultants, Inc. Seattle, WA, USA

ABSTRACT

Make advanced spam filtering work with your existing mail tools.

CITED BY

Pawel Gburzynski , Jacek Maitan, Fighting the spam wars: A remailer approach with restrictive aliasing, ACM Transactions on Internet Technology (TOIT), v.4 n.1, p.1-30, February 2004

INDEX TERMS

Primary Classification:
H. Information Systems
↪ **H.4** INFORMATION SYSTEMS APPLICATIONS
↪ **H.4.3** Communications Applications
↪ **Subjects:** Electronic mail

Additional Classification:
H. Information Systems
↪ **H.5** INFORMATION INTERFACES AND PRESENTATION (I.7)
↪ **H.5.2** User Interfaces (D.2.2, H.1.2, I.3.6)
↪ **Subjects:** Training, help, and documentation

General Terms:
Algorithms, Security

An Introduction to the Spambayes Project

A trainable system that works with your current e-mail system to catch and filter junk mail.

by Richie Hindle

The Spambayes Project is one of many projects inspired by Paul Graham's ``A Plan for Spam" (www.paulgraham.com/spam.html). This famous article talks about using a statistical technique called Bayesian Analysis to identify whether an e-mail message is spam. For the full story of how the mathematics behind Spambayes works and how it has evolved, see Gary Robinson's accompanying article on page 58.

In a nutshell, the system is trained by a set of known spam messages and set of known non-spam, or ``ham", messages. It breaks the messages into tokens (words, loosely speaking) and gives each token a score according to how frequently it appears in each type of message. These scores are stored in a database. A new message is tokenized and the tokens are compared with those in the score database in order to classify the message. The tokens together give an overall score--a probability that the message is spam.

The fact that you train Spambayes by using your own messages is one of its strengths. It learns about the kinds of messages, both ham and spam, that you receive. Other spam-filtering tools that use blacklists, generic spam-identification rules or databases of known spams don't have this advantage.

The Spambayes software classifies e-mail by adding an X-Spambayes-Classification header to each message. This header has a value of spam, ham or unsure. You then use your existing e-mail software to filter based on the value of that header. We use a scale of spamminess going from 0 (ham) to 1 (spam). By default, < 0.2 means ham and > 0.9 means spam. Any e-mail between those figures is marked as unsure. You can tune these thresholds yourself; see below for information on how to configure the software.

Why Spambayes Is Different

Spambayes is different from other spam classifiers in three ways: its test-based design philosophy, its tokenizer and its classifier.

We can all think of obvious ways to identify spam: it has SHOUTING subject lines; it tells you how to Make Money Fast!!!; it purports to be from the vice president of Nigeria or his wife. It's tempting to tune any spam-classification software according to obvious rules. For instance, it should obviously be case-sensitive, because FREE is a much better spam clue than free. But the Spambayes team refused from the outset to take anything at face value. One of the earliest components of the software was a solid testing framework, which would compare new ideas against the previous version. Any idea that didn't improve the results was ditched. The results were often surprising; for instance, case sensitivity made no significant difference. This prove-it-or-lose-it approach has helped develop an incredibly accurate system, with little wasted effort.

The tokenizer does the job of splitting messages into tokens. It has evolved from simple split-on-whitespace into something that knows about the structure of messages, for instance, tagging words in the Subject line so that they are separately identified from words in the body. It also knows about their content, for instance, tokenizing embedded URLs differently from plain text. All the special rules in the tokenizer have been rigorously tested and proven to improve accuracy. This includes deliberately hiding certain tokens--for example, we strip HTML decorations and ignore most headers by default. Surprising decisions, but they're backed up by testing.

The classifier is the statistical core of Spambayes, the number cruncher. This has evolved a great deal since its beginnings in Paul Graham's article, again through test-based development. Gary's article, ``A Statistical Approach to the Spam Problem" (page 58), covers the classifier in detail.

Requirements and Installation

The Spambayes software is available for download from sf.net/projects/spambayes. It requires Python 2.2 or above and version 2.4.3 or above of the Python e-mail package. If you're running Python 2.2.2 or above, you should already have this. If not, you can download it from mimelib.sf.net and install it: unpack the archive, `cd` to the email-2.4.3 directory and type `setup.py install`. This will install it in your Python site-packages directory. You'll also need to move aside the standard e-mail library; go to your Python Lib directory, and rename the file email as email_old.

Keeping up to Date

Because the project is in constant development, things are sure to change between my writing this article and the magazine hitting the newsstand. I'll publish a summary of any major changes on an Update page at www.entrian.com/spambayes.

Some of the things we're working on as I write this article include more flexible command-line training; enabling integration with more e-mail clients, such as Mutt; web-based configuration; security features for the web interface; and easier installation. I'll provide full details of these items on the Update page.

Components

Three classifier programs are in the Spambayes software: a procmail filter, a POP3 proxy and a plugin for Microsoft Outlook 2000. I cover the procmail filter and the POP3 proxy in this article. A web interface (covered below) and various command-line utilities, test harnesses and so on are also part of Spambayes; see the documentation that comes with the software for full details.

Procmail-Based Setup

If you use a procmail-based e-mail system, this is how the Spambayes procmail system works:

- All your existing mail has a new X-Spambayes-Trained header. The software uses this to keep track of which messages it has already learned about.
- The software looks at all your incoming mail. Messages it thinks are spam are put in a ``spam" mail folder. Everything else is delivered normally.
- Every morning, it goes through your mail folders and trains itself on any new messages. It also picks up mail that's been refiled--something it thought was ham but was actually spam and vice versa. Be sure to keep spam in your spam folder for at least a day or two before deleting it. We suggest keeping a few hundred messages, in case you need to retrain the software.

You'll need a working crond to set up the daily training job. Optionally, you can have a mailbox of spam and a mailbox of ham to do some initial training.

To set up Spambayes on your procmail system, begin by installing the software. I'll assume you've put it in $HOME/src/spambayes. Then, create a new database:

```
$HOME/src/spambayes/hammiefilter.py -n
```

If you exercise the option to train Spambayes on your existing mail, type:

```
$HOME/src/spambayes/mboxtrain.py \
-d $HOME/.hammiedb -g $HOME/Mail/inbox \
-s $HOME/Mail/spam
```

You can add additional folder names if you like, using -g for good mail folders and -s for spam folders. Next, you need to add the following two recipes to the top of your .procmailrc file:

```
:0fw
| $HOME/src/spambayes/hammiefilter.py

:0
* ^X-Spambayes-Classification: spam
$HOME/Maildir/.spam/
```

The previous recipe is for the Maildir message format. If you need mbox (the default on many systems) or MH, the second recipe should look something like this:

```
:0:
* ^X-Spambayes-Classification: spam
$HOME/Mail/spam
```

If you're not sure what format you should use, ask your system administrator. If you are the system administrator, check the documentation of your mail program. Most modern mail programs can handle both Maildir and mbox.

Using `crontab -e`, add the following cron job to train Spambayes on new or refiled messages every morning at 2:21 AM:

```
21 2 * * * $HOME/src/spambayes/mboxtrain.py -d
$HOME/.hammiedb -g $HOME/Mail/inbox
-s $HOME/Mail/spam
```

You also can add additional folder names here. It's important to do this if you regularly file mail in different folders; otherwise Spambayes never learns anything about those messages.

Spambayes should now be filtering all your mail and training itself on your mailboxes. But occasionally a message is misfiled. Simply move that message to the correct folder, and Spambayes learns from its mistake the next morning.

Many thanks to Neale Pickett for the information in this section.

Setting Up the POP3 Proxy and the Web Interface

If you don't use Procmail or don't want to mess with it, or if you want to set up the software on a non-UNIX machine, you can use the POP3 proxy. This is a middleman that sits between your POP3 server and your e-mail program, and it adds an X-Spambayes-Classification header to e-mails as you retrieve them. You also can use the POP3 proxy with Fetchmail; simply reconfigure Fetchmail to talk to the POP proxy rather than your real POP3 server.

The web interface lets you pretrain the system, classify messages and train on messages received via the POP3 proxy, all through your web browser. The software is configured through a file called bayescustomize.ini. This is true of the Procmail filter as well. There's no need to change any of the defaults to use it out-of-the-box, but the POP3 proxy needs to be set up with the details of your POP3 server. All the available options and their defaults live in a file called Options.py, but you need to look at that only if you're terminally curious or want to do advanced tuning. The minimum you need to do is create a bayescustomize.ini file like this:

```
[pop3proxy]
pop3proxy_servers: pop3.example.com
```

where *pop3.example.com* is wherever you currently have your e-mail client configured to collect mail. The proxy runs on port 110 by default. This is fine on non-UNIX platforms, but on UNIX you'll want to use a different one by adding this line:

```
pop3proxy_ports: 1110
```

to the [pop3proxy] section of bayescustomize.ini. If you collect mail from more than one POP3 server, you can provide a list of comma-separated addresses in pop3proxy_servers and a corresponding list of comma-separated port numbers in pop3proxy_ports. Each port proxies to the corresponding POP3 server.

You can now run pop3proxy.py. This prints some status messages, which should include something like:

```
Listener on port 1110 is
    proxying pop3.example.com:110
User interface url is http://localhost:8880
```

This means the proxy is ready for your e-mail client to connect to it on port 1110, and the web interface is ready for you to point your browser at the given URL. To access the web interface from a different machine, replace localhost with the name of the machine running pop3proxy.py.

Classifying Your E-mail Using the POP3 Proxy

You now need to configure your e-mail client to collect mail from the proxy rather than from your POP3 server. Where you currently have pop3.example.com, port 110, set up as your POP3 server, you need to set it to localhost, port 1110. If you're running the proxy on a different machine from your e-mail client, use *machinename*, port 1110.

Classifying your mail is now as easy as clicking ``Get new mail''. The proxy adds an X-Spambayes-Classification header to each message, and you can set up a filter in your mail program to file away suspected spam in its own folder. Until you do some training, however, all your messages are classified as unsure.

Once you're up and running, you should check your suspected spam folder periodically to see whether any real messages slip through, so-called false positives. As you train the system, this will happen less and less often.

Training through the Web Interface

Initial training isn't an absolute requirement, but you'll get better results from the outset if you do it. You can use the upload a message or mbox file form to train via the web interface, either on individual messages or UNIX mbox files.

Once you're up and running, you can use the web interface to train the system on the messages the POP3 proxy has seen. The Review messages page lists your messages, classified according to whether the software thought they were spam, ham or unsure. You can correct any mistakes by checking the boxes and then clicking Train. After a couple of days (depending on how much e-mail you get), there'll be very few mistakes.

Figure 1. Spambayes Proxy Web Training Page

Training Tips

Spambayes does an excellent job of classifying your mail, but it's only as good as the data on which you train it. Here are some tips to help you get the best results:

- Don't train on old mail. The characteristics of your e-mail change over time--sometimes subtly, sometimes dramatically--so it's best to use recent mail.
- Take care when training. If you mistakenly train a spam message as ham, or vice versa, it will throw off the classifier.
- Try to train on roughly as much spam as ham. This isn't critical, but you'll get better results with a fair balance.

Possible Future Directions

The Spambayes software is in constant development. Many people are involved, and we have many ideas about what to do next. Here's a taste of where the project might go:

- Improving the tokenizer and classifier as new research reveals more accurate ways to classify spam.
- Intelligent autotraining: once the system is up and running, it should be possible for it to keep itself up-to-date by training itself, with users correcting only the odd mistake. We're already doing something along these lines with the Procmail system, but we're looking at ways of making it more automated and compatible with all platforms.
- SMTP proxy: to train the system from any e-mail client on any platform, you could send a message to a special ham or spam address. This could be a simple way to correct classification mistakes, and it would combine well with intelligent auto-training techniques.
- Database reduction: the more you train the system, the larger its database gets. We're looking at

ways to keep the database size down.

- Integration with spam-reporting tools: the web interface and the e-mail plugins could let you report spams to systems like Vipul's Razor and Pyzor.
- More e-mail client integration: we already have the Outlook plugin, and we'd like to integrate with more e-mail clients. The POP3 proxy and the web interface work well with any e-mail client, but having a Delete as Spam button right there in your e-mail client is much more convenient than switching to your web browser.
- Better documentation: we aim to publish documentation on how to set up Spambayes on all the popular platforms and e-mail clients.

By the time this article is in print, some of these things already may be happening; see my Update page at www.entrian.com/spambayes for details.

Richie Hindle *is a professional software engineer in the UK. He works full-time writing business intelligence software, and in his spare time he works on Spambayes and his own Python projects at www.entrian.com. He only occasionally wears a silly hat.*